

Data Processing Workflows in the Social Sciences: Representation and Automatic Generation (Abstract)

José Luis Ambite
Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292
ambite@isi.edu

Dipsy Kapoor
Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292
dipsy@isi.edu

Mountu Jinwala
Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292
jinwala@isi.edu

1. INTRODUCTION

Much of the work of social scientists and government practitioners is consumed by accessing, collating, and analyzing data. This is particularly true in the planning and economic modeling agencies. Unfortunately, there is a severe lack of tools to facilitate this process and much of the integration is done manually by ad-hoc methods. Moreover, raw data are of limited utility. Usually these data are the input to models of more complex phenomena that produce additional data of interest. For example, in our commodity flow domain, we derive truck traffic along specific highway links within a metropolitan area, based on quite far-removed raw (source) data such as employment, imports into and exports out of the region, etc. by using a complex workflow of operations.

The goal of the Argos project is to improve this state of affairs by providing a framework to (1) describe data products and data processing operations so that they can be shared and reused and (2) automatically generate new data products on demand by automatically composing data processing workflows using available sources and operations. In this abstract, we outline our solutions to these two challenges. A novel contribution is that our approach can automatically insert operations that make the inputs and outputs of different operations compatible (so called *shims* [1]).

2. DOMAIN REPRESENTATION

One of the major challenges to automating computational workflows is understanding the data products present in available sources and operations. After some effort, a domain expert can accurately describe such data. However, such descriptions are rarely recorded and much less formalized unambiguously. To ensure a clear data semantics, we have developed a formal ontology for our goods movement domain.

The Argos Ontology represents the concepts and relations of our transportation domain in the expressive Powerloom language, a first-order logic with recursion [3]. Powerloom provides logical inference services to the Argos system, in particular, *subsumption*. Subsumption proves that membership in a class (or relation) logically implies membership in another class (or relation). First-order logical inference is undecidable, hence Powerloom is incomplete. Nevertheless, Powerloom is specially optimized to compute subsumption and Powerloom proves the inferences required by our (quite expressive) ontology efficiently.

Figure 1 shows some sample concept, instance and rule definitions of the Argos ontology. The `Flow` concept rep-

resents a transfer of a product between two geospatial areas (from an origin to a destination) using a transportation mode measured during a particular time interval. For example, an instance of `Flow` is the domestic exports by air (a `TransportationMode`) of Pharmaceutical and Chemical products from the Los Angeles-Riverside-Orange County, CA, Consolidated Metropolitan Statistical Area (LACMSA) in 2000, which amounts to 2226 million US dollars.

The ontology also encodes information about well-known entities in the domain. For example, Figure 1 shows the fact that Los Angeles County (`g-LA`) is geographically contained in (is a `geoPartOf`) the LACMSA area, as well as Ventura County (`g-VT`), something not immediately apparent from the LACMSA name. Finally, the ontology includes rules clarifying the semantics of the concepts and relations. For example, the *recursive* rule in Figure 1 specifies the transitivity of geospatial containment (`geoPartOf`).

```
(DEFCONCEPT Flow (?x) :<=>           ;; concept definition
  (exists (?o ?d ?p ?t ?u ?m ?v)
    (AND (Data ?x)
      (hasOrigin ?x ?o) (Geo ?o)
      (hasDestination ?x ?d) (Geo ?d)
      (hasProduct ?x ?p) (Product ?p)
      (hasTimeInterval ?x ?t) (TimeInterval ?t)
      (hasUnit ?x ?u) (Unit ?u)
      (hasMode ?x ?m) (TransportationMode ?m)
      (hasValue ?x ?v) (Number ?v) )))

(USGeo g-LACMSA) (USCounty g-LA)      ;; instance assertions
(geoPartOf g-LA g-LACMSA) (geoPartOf g-VT g-LACMSA)

(forall (?x ?z)                          ;; inference rule
  (=> (exists (?y)
      (and (geoPartof ?x ?y) (geoPartof ?y ?z)))
    (geoPartof ?x ?z)))
```

Figure 1: Argos Ontology: Sample Definitions

Using this formal ontology we describe the data products provided by sources, and required or computed by data processing operations. For example, Figure 2 shows the description of the contents of a table that provides the number of jobs in 2000 for each Traffic Analysis Zone (TAZ) contained in the LACMSA, for products categorized following the 1999 Standard Industrial Classification (SIC) codes (with a granularity of 4 digits). Since the data description uses the logical biconditional (`<=>`), it means that the table has the *complete* set of tuples that satisfy the relation definition, i.e., the table contains values for *all* the products of type `Product-sic-4-1999` for *all* the TAZs in the LACMSA.

```

(defrelation Data-Rel-Employment-2000-LACMSA-TAZ-SIC
  ((?county USCounty) (?jobs Number)
   (?p Product-sic-4-1999)(?taz TAZ)) :<=>
  (exists (?o)
   (AND (Measurement ?o)
        (hasProduct ?o ?p)
        (hasGeo ?o ?taz) (geoPartof ?taz ?county)
                          (geoPartof ?county g-LACMSA)
        (hasUnit ?o u-NumberOfJobs)
        (hasTimeInterval ?o 2000)
        (hasValue ?o ?jobs))))

```

Figure 2: Sample Data Product Description

3. AUTOMATICALLY COMPOSING DATA PROCESSING WORKFLOWS

We assume that sources and operations have been developed independently. For example, the operations may be web services and the sources external databases. This presents two challenges. First, each source may use a different schema. Second, the data produced by a source or operation may not be input directly into other operations, but need some kind of transformation.

Argos addresses both these challenges. First, it resolves the semantic heterogeneity by mapping all data products to a common ontology. Second, Argos provides a library of domain-independent operations and a framework to define generic domain-dependent operations that can bridge the differences between the input required by some operation and the output provided by another (*shims* [1]).

3.1 Operations

A data processing operation is represented by its input/output signature. Each input or output is described by a relation definition in the ontology (e.g., Figure 2). An operation can have multiple inputs and outputs. There are three types of operations supported in Argos:

Sources and Domain-Dependent Operations. The inputs and outputs are described by data relations predefined in the ontology. A source is a domain-dependent operation that requires no inputs.

Domain-Independent Operations. In order to bridge the inputs and outputs of different operations, Argos provides a set of built-in domain-independent operations similar to the relation algebra operations: selection, projection, join, and union. However, the system uses the background ontology to prove that inserting such operation is semantically valid. We illustrate the process with selection. The other operators are analogous.

Assume that the planner wants to obtain the employment data for the TAZs in Los Angeles County, but the available operators can only produce the employment data for all the TAZs of the LACMSA (cf. Figure 2). Using the ontology, the system reasons that since Los Angeles County is geographically contained in the LACMSA (cf. Figure 1) the desired relation is a subset of the available one. After also checking that county is an output attribute of the provider relation, the system will insert a selection operator.

Generic Domain-Dependent Operations. There are a variety of operators that lie between the completely domain-specific operators (described by predefined datasets), and the (relational-algebra-like) domain-independent operators. Product conversion is a prime example of a generic, but domain-dependent operator.

Economic data is reported in a variety of product/industry classifications (NAICS, SCTG, SIC, ...). Our project social scientists have created translation tables between several of these classifications. Thus, we added a generic product conversion operator to the Argos library. To satisfy a data request for products in a classification C2, this operator subgoals on obtaining the data in classification C1 and a translation table from C1 to C2.

3.2 Planning

The Argos planner automatically generates a workflow of sources and operations in response to an user data request. Our planner performs a regression search in plan space (cf. [2]), starting with the user data request as goal, until it finds a plan that computes such request using available operators where all data inputs to operators can be satisfied by other operators or by the available sources.

The basic plan refinement step is to satisfy an operator input with the output of another operator or source. In order to ensure that the input and the output data relations are semantically compatible, the planner performs an equivalence test, that is, it checks subsumption in both directions.

4. DISCUSSION

The Argos planner and executor are fully implemented. We have developed a core ontology for our transportation domain with about 40 concepts and 10 relations. We also described 17 sources and 11 domain-specific operations.

Our initial experiments are promising. The planner generates a workflow with 17 operations (7 sources, 4 domain-specific operations and 6 product conversions generated on-the-fly) in about 20 seconds. A larger workflow with 54 operations (17 sources, 11 domain-specific operators, 8 product conversions, and 18 projection operators) takes 2 minutes and 44 seconds.

In the immediate future, we plan to scale the number of operators and sources in our commodity flow domain. In addition, we want to broaden the scope of work to other questions of spatial urban structure.

5. ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Award No. EIA-0306905. We would like to thank the social scientists of the Argos project for their contributions: Genevieve Giuliano, Peter Gordon, Qisheng Pan, LanLan Wang, and JiYoung Park; as well as our government partner agencies.

6. REFERENCES

- [1] D. Hull, R. Stevens, P. Lord, C. Wroe, and C. Goble. Treating shimantic web syndrome with ontologies. In *Procs. 1st AKT workshop on Semantic Web Services (AKT-SWS04)*, Milton Keynes, UK, 2004.
- [2] C. A. Knoblock. Building a planner for information gathering: A report from the trenches. In *Procs. 3rd International Conference on Artificial Intelligence Planning Systems*, Edinburgh, Scotland, 1996.
- [3] R. MacGregor. A description classifier for the predicate calculus. In *Procs. 12th National Conference on Artificial Intelligence*, Seattle, WA, 1994.